

# NURBS-SEM: a hybrid spectral element method on NURBS maps for the solution of elliptic PDEs on surfaces

Giuseppe Pitton<sup>a</sup>, Luca Heltai<sup>a,\*</sup>

<sup>a</sup>*SISSA, Via Bonomea 265, Trieste, Italy*

---

## Abstract

Non Uniform Rational B-spline (NURBS) patches are a standard way to describe complex geometries in Computer Aided Design tools, and have gained a lot of popularity in recent years also for the approximation of partial differential equations, via the Isogeometric Analysis (IGA) paradigm. However, spectral accuracy in IGA is limited to relatively small NURBS patch degrees (roughly  $p \leq 8$ ), since local condition numbers grow very rapidly for higher degrees. On the other hand, traditional Spectral Element Methods (SEM) guarantee spectral accuracy but often require complex and expensive meshing techniques, like transfinite mapping, that result anyway in *inexact* geometries. In this work we propose a hybrid NURBS-SEM approximation method that achieves spectral accuracy and maintains exact geometry representation by combining the advantages of IGA and SEM.

As a prototypical problem on non trivial geometries, we consider the Laplace–Beltrami and Allen–Cahn equations on a surface. On these problems, we present a comparison of several instances of NURBS-SEM with the standard Galerkin and Collocation Isogeometric Analysis (IGA).

**Keywords:** NURBS, Spectral Element Methods, Collocation Methods, Isogeometric Analysis, Laplace Beltrami, Allen Cahn

---

## 1. Introduction

Isogeometric analysis (IGA) is a numerical approximation paradigm for Partial Differential Equations (PDE) based on the exploitation of the same Non Uniform Rational B-spline (NURBS) basis functions used in Computer Aided Design (CAD) tools, and it has become a standard practice since its introduction in [26, 15].

---

\*Corresponding author

*Email addresses:* `gpitton@sisssa.it` (Giuseppe Pitton), `luca.heltai@sisssa.it` (Luca Heltai)

It can be argued that the main merit of IGA is the removal of the mesh generation process required by other standard approximation methods (FEM, FVM, ...) and the resulting tight interaction between CAD and analysis tools. Such a tight connection is particularly significant when the problem is formulated directly on lower dimensional manifolds embedded in the physical space, i.e., on surfaces in three dimensional space or curves in two or three-dimensional spaces, since in this case the Boundary Representation (BREP) standard which is common to all major CAD tools can be used *as is*, without the need to create volumetric NURBS representations. This is the case for plates and shell structures [29, 30], Reissner–Mindlin shells [9, 22, 21, 43], boundary integral formulations [25, 24], or when the physical problem of interest is naturally defined on manifolds [5, 6, 18] embedded in a Euclidean space.

Standard Finite Element approximation of such PDEs typically resort to an approximation of the geometry by means of piecewise polynomial patches. This results in an approximation of the curvature of surfaces that may significantly affect the approximation quality. On the other hand, most of the geometries of practical interest can be represented exactly by B-splines or NURBS [36], making IGA an ideal candidate for these types of problems.

NURBS-based Isogeometric analysis is a higher-order approach that allows an alternative refinement strategy with respect to the standard  $h$ - and  $p$ -refinements used in finite element analysis: the so called  $k$ -refinement [26], where the regularity of the basis functions is raised at each refinement stage. This is of particular importance for eigenvalue problems, where IGA provides good approximations of a much larger part of the spectrum than standard FEA, as shown by [16] in a structural analysis context. The accuracy and usability of  $k$ -refinement, however, is limited by cross-patch regularity. For multiple-patch geometries, the global condition number of the system matrices presents the same behaviour under  $p$ - and  $k$ -refinement. The analysis of these methods [8, 7] proves that spectral accuracy can be achieved, but the error constants depend exponentially on the degree of the NURBS basis functions. In practical applications this obstructs IGA from reaching spectral accuracy, which proves to be very efficient for relatively low degrees, and deteriorates when the basis functions' degree exceeds 8 or 9.

In this paper we present an alternative approximation strategy inspired by the Spectral Element Method (SEM), that aims at combining the advantages of IGA and SEM, similarly to what is done in [41] between FEM and NURBS. This approach is particularly efficient for those cases where it is important to treat the geometry exactly, but it would be desirable to achieve spectral accuracy on the solution of the PDE. This is possible by breaking the iso-parametric paradigm of standard SEM and IGA, and constructing the push-forward of SEM basis functions through the NURBS description of the geometry.

This article is organized as follows. In Section 2, we briefly introduce the problem of solving an elliptic, second order PDE on a surface, and the Galerkin and Collocation strategies are introduced. In Section 3, we give a detailed description of the Spectral Element schemes introduced in this work, as well as a unified description encompassing some known methods such as IGA. The

algebraic details leading to an efficient implementation of the Spectral Element Methods are also introduced in Section 3, followed by a discussion on the implementation of essential and mixed boundary conditions. In Section 4, we assess the performance of several hybrid NURBS-SEM instances by making a comparison with standard Galerkin and collocation Isogeometric Analysis for the approximate solution of the Laplace–Beltrami and Allen–Cahn equations on a surface, and discuss the results. The Conclusions in Section 5 close the paper.

## 2. Laplace–Beltrami and related Equations

The Laplace–Beltrami operator can be thought of as an extension of the Laplace operator from a domain in  $\mathbb{R}^n$  to a manifold. We briefly outline its derivation on a general Riemannian Manifold, together with its weak formulation. The subject is classical, and we refer for instance to [33, 31] for an introduction to Riemannian Geometry, and to [32], [37] for a detailed treatment on the properties of the Laplace–Beltrami operator (notably, its spectral properties).

Consider a bounded, orientable surface  $\Omega$  embedded in  $\mathbb{R}^3$ . Any regular surface can be described as a Riemannian Manifold, with metric tensor  $g$ . We assume that for each point  $p \in \Omega$ , there exists a local coordinate description of  $\Omega$  that consists of a pair  $(U, \mathbf{x}^{-1})$ , where  $U$  is a neighbourhood of  $p$  homeomorphic to an open set of  $\mathbb{R}_+^2$  (namely, the upper half plane) and  $\mathbf{x} : \mathbb{R}^2 \rightarrow U$  is a differentiable homeomorphism. Since we are mainly concerned with CAD applications, we consider the special case where  $\mathbf{x}$  is a tensor product of NURBS functions, that will be described in detail in Section 3.1.

In this representation, the local coordinates are typically given as maps from the reference domain  $\hat{\Omega}$ , that we take as the unit square, to the Euclidean space  $\mathbb{R}^3$ , i.e.,  $\mathbf{s} = s^1 \mathbf{e}_1 + s^2 \mathbf{e}_2 \in \hat{\Omega}$ , and the surface is described through the coordinates of its points  $\mathbf{x}(\mathbf{s}) = x^i(\mathbf{s}) \mathbf{e}_i \in \mathbb{R}^3$  as  $\mathbf{s}$  varies in  $\hat{\Omega}$ .

A basis for the tangent space at a point is given by the partial derivatives of  $\mathbf{x}$  with respect to the reference coordinates:

$$\mathbf{g}_\alpha(\mathbf{s}) := \frac{\partial x^i}{\partial s^\alpha}(\mathbf{s}) \mathbf{e}_i, \quad (1)$$

where the summation with respect to  $i$  is implied. Here and in the following, we adopt Einstein’s summation convention on repeated indices. The metric tensor has the following local representation:

$$g_{\alpha\beta} := \mathbf{g}_\alpha \cdot \mathbf{g}_\beta = \mathbf{x}_{,\alpha} \cdot \mathbf{x}_{,\beta}, \quad (2)$$

while the components of the inverse metric tensor  $g^{\alpha\beta}$  are obtained by inversion of the  $2 \times 2$  matrix  $g_{\alpha\beta}$ . We adopt the Einstein summation convention, with greek indices to indicate coordinates in  $\hat{\Omega} \subset \mathbb{R}^2$ , and latin indices for coordinates in  $\mathbb{R}^3$ .

Any Riemannian Manifold is naturally endowed with a Levi–Civita connection  $\nabla$ , that introduces a precise definition for the covariant derivatives of scalar functions or vector fields on a manifold. The surface gradient of a scalar function

$f$  at a point on  $\Omega$  is given by a differential form, whose representation in the local coordinate system is:

$$\nabla f = \nabla_\mu f \mathbf{g}^\mu := \partial_\mu f \mathbf{g}^\mu = \frac{\partial f}{\partial s^\mu} \mathbf{g}^\mu, \quad (3)$$

where  $\{\mathbf{g}^\nu = g^{\nu\mu} \mathbf{g}_\mu\}$  is the dual basis to  $\{\mathbf{g}_\mu\}$ , i.e.,  $\mathbf{g}^\mu \cdot \mathbf{g}_\nu = \delta^\mu_\nu$  with  $\delta$ , the Kronecker delta, equal to one if  $\nu = \mu$ , and zero otherwise. A representation of  $\nabla f$  in the tangent space  $T\Omega$  is the vector field obtained by raising indices:

$$\nabla f = \nabla^\mu f \mathbf{g}_\mu = g^{\mu\nu} \frac{\partial f}{\partial s^\nu} \mathbf{g}_\mu. \quad (4)$$

For a vector field  $\mathbf{u} \in T\Omega$ , the covariant derivative is defined as:

$$\nabla \mathbf{u} := \nabla(u^\alpha \mathbf{g}_\alpha) = \left( \frac{\partial u^\alpha}{\partial s^\beta} + \Gamma^\alpha_{\beta\gamma} u^\gamma \right) \mathbf{g}_\alpha \otimes \mathbf{g}^\beta, \quad (5)$$

where  $\Gamma^\mu_{\alpha\beta}$  are the Christoffel symbols of the second kind, that can be expressed in terms of partial derivatives of the metric tensor, as follows. First, the Christoffel symbols of the first kind are defined:

$$\Gamma_{\lambda\mu\nu} = \frac{1}{2} (\partial_\nu g_{\lambda\mu} + \partial_\mu g_{\lambda\nu} - \partial_\lambda g_{\mu\nu}), \quad (6)$$

then, the first index is raised:

$$\Gamma^\lambda_{\mu\nu} = g^{\lambda\varrho} \Gamma_{\varrho\mu\nu}. \quad (7)$$

With these notions, it is possible to introduce the Laplace–Beltrami operator acting on a scalar function as:

$$\begin{aligned} \Delta_B &:= \nabla^\alpha \nabla_\alpha = g^{\alpha\beta} \nabla_\beta \nabla_\alpha = g^{\alpha\beta} \nabla_\beta (\partial_\alpha) \\ &= g^{\alpha\beta} \left[ \partial_{\alpha\beta}^2 + \Gamma^\mu_{\alpha\beta} \partial_\mu \right], \end{aligned} \quad (8)$$

The derivatives of a function  $u(\mathbf{x})$  defined on the surface can be obtained explicitly by the chain rule:

$$\partial_\alpha u(\mathbf{x}(\mathbf{s})) = \frac{\partial u}{\partial x^i} \Big|_{\mathbf{x}(\mathbf{s})} \frac{\partial x^i}{\partial s^\alpha} \Big|_{\mathbf{s}}. \quad (9)$$

From this point on, with some abuse of notation we will identify the surface  $\Omega$  with its parametrization  $\mathbf{x}(\mathbf{s})$ .

### 2.1. Laplace–Beltrami Equation

Let us now define the Laplace–Beltrami equation on a surface with boundary  $\Omega$ . Let  $\mathcal{C}^1(\Omega)$  be the set of differentiable functions on  $\Omega$  with values in  $\mathbb{R}$ , and  $\mathcal{C}^2(\Omega)$  the set of functions on  $\Omega$  with continuous second derivative, then

the Laplace–Beltrami problem is: given a function  $f \in \mathcal{C}^0(\Omega)$ , find a function  $u \in \mathcal{C}^2(\Omega)$  such that:

$$\begin{cases} -\Delta_B u = f & \text{in } \Omega \\ u = h_D & \text{on } \partial_D \Omega \\ \partial_n u = h_N & \text{on } \partial_N \Omega, \end{cases} \quad (10)$$

where  $\partial_D \Omega$  and  $\partial_N \Omega$  form a partition of the boundary  $\partial \Omega$  of  $\Omega$ , and  $h_D : \partial_D \Omega \rightarrow \mathbb{R}$  and  $h_N : \partial_N \Omega \rightarrow \mathbb{R}$  are respectively the given Dirichlet and Neumann boundary data. For surfaces without boundary, the treatment is conceptually similar, except that boundary conditions cannot be imposed, and instead the average of  $u$ , or its value at a point must be specified.

In the following, we refer mainly to the variational form of Equation (10). Let  $f \in H^{-1}(\Omega)$ ,  $h_D \in H^{\frac{1}{2}}(\partial_D \Omega)$  and  $h_N \in H^{-\frac{1}{2}}(\partial_N \Omega)$  be given. We seek for a solution  $u \in H_D^1(\Omega)$  such that:

$$(\nabla v, \nabla u) = \langle\langle v, h_N \rangle\rangle + \langle v, f \rangle \quad \forall v \in H_0^1(\Omega), \quad (11)$$

where the round brackets denote the  $L^2(\Omega)$  inner product, the brackets  $\langle\langle \cdot, \cdot \rangle\rangle$  are used to express the duality pairing between the Sobolev spaces  $H^{-\frac{1}{2}}(\partial_N \Omega)$  and  $H^{\frac{1}{2}}(\partial_N \Omega)$ , while the brackets  $\langle \cdot, \cdot \rangle$  are used for the duality pairing between the Sobolev spaces  $H^{-1}(\Omega)$ , and  $H_D^1(\Omega)$  where

$$H_D^1 := \{v \in H^1(\Omega) : \gamma_D u = h_D\}, \quad (12)$$

for a suitable trace operator  $\gamma_D : H_D^1(\Omega) \rightarrow H^{\frac{1}{2}}(\partial_D \Omega)$ .

Approximation methods for the variational problem (11) based on the Petrov–Galerkin projection stem from the introduction of two finite dimensional spaces  $V^n, W^n$ . An approximation  $u^n \in V^n$  is defined as the solution to:

$$(\nabla v, \nabla u^n) = \langle\langle v, h_N \rangle\rangle + \langle v, f \rangle \quad \forall v \in W^n. \quad (13)$$

Let  $\{\phi_j\}_{j=1}^n$  and  $\{\psi_i\}_{i=1}^n$  be two sets of basis functions respectively for  $V^n$  and  $W^n$ . Then, the approximate solution can be expressed as  $u^n(\mathbf{x}) = \sum_{j=1}^n u_j^n \phi_j(\mathbf{x})$ , and replacing this expansion in Equation (13) and choosing as test functions  $v = \psi_i$  from a set of basis functions for  $W^n$ , Equation (13) is equivalent to the following linear system:

$$\sum_{j=1}^n u_j^n (\nabla \psi_i, \nabla \phi_j) = \langle\langle \psi_i, h_N \rangle\rangle + \langle \psi_i, f \rangle \quad \forall \psi_i, i = 1, \dots, n. \quad (14)$$

Petrov–Galerkin schemes allow as well to construct approximations for nonlinear Laplace–Beltrami analogues. We discuss the nonlinear case by making the example of the Allen–Cahn equation:

$$-\Delta_B u - u + u^3 = f \quad (15)$$

with the same boundary conditions discussed after Equation (10). Before applying the Petrov–Galerkin projection, it is convenient to set up an iterative solution scheme. Here, we consider the following fixed-point iteration:

1. an initial guess  $u_0 \in V^n$  is found by solving the linear part of the equation:

$$-\Delta_B u_0 - u_0 = f; \quad (16)$$

2. at the  $n$ -th iteration, a new approximation  $u_{n+1} \in V^n$  is obtained by solving the linear equation:

$$-\Delta_B u_{n+1} - u_{n+1} + u_n^2 u_{n+1} = f, \quad (17)$$

whose weak form is: find  $u_{n+1} \in V^n$  s.t.:

$$(\nabla v, \nabla u_{n+1}) - (v, u_{n+1}) + (v, u_n^2 u_{n+1}) = \langle\langle v, h_N \rangle\rangle + \langle v, f \rangle \quad \forall v \in W^n. \quad (18)$$

3. to judge on the quality of the current approximation  $u_{n+1}$ , it is convenient to consider the increment:

$$d_n = \|u_{n+1} - u_n\|; \quad (19)$$

the iterative scheme is halted whenever  $d_n$  is sufficiently small, e.g. by requiring that its order of magnitude be comparable with the machine epsilon, the smallest number representable in floating point arithmetic.

## 2.2. Algebraic problem

The following variational forms are naturally associated with the Laplace–Beltrami problems described in Section 2.1:

$$\mathcal{A}(v, u) = (\nabla v, \nabla u), \quad \mathcal{F}(v) = \langle\langle v, h_N \rangle\rangle + \langle v, f \rangle, \quad \mathcal{N}(v, w, u) = (v, w^2 u) - (v, u). \quad (20)$$

The Laplace–Beltrami Equation (13) can be written in terms of these differential forms as:

$$\mathcal{A}(v, u) = \mathcal{F}(v) \quad \forall v \in W^n, \quad (21)$$

and the  $n$ -th iteration of the fixed-point scheme for the Allen–Cahn problem as:

$$\mathcal{A}(v, u_{n+1}) + \mathcal{N}(v, u_n, u_{n+1}) = \mathcal{F}(v) \quad \forall v \in W^n. \quad (22)$$

Evaluating the differential forms defined in Equation (20) on the basis sets  $\{\phi_j\}, \{\psi_i\}$  yields the matrices:

$$K_{ij} = \mathcal{A}(\psi_i, \phi_j) \quad (23)$$

for Laplace–Beltrami, and

$$K_{ij} = \mathcal{A}(\psi_i, \phi_j) + \mathcal{N}(\psi_i, u_n, \phi_j) \quad (24)$$

for a step of the Allen–Cahn fixed-point iteration. Note that in this latter case the matrix  $K$  is not symmetric. Associated with  $\mathcal{F}$ , the following column vector arises:

$$\mathbf{f}_i = \mathcal{F}(\psi_i), \quad (25)$$

and it is also useful to introduce the column vector  $\mathbf{u}$ , whose components are the solutions' expansion coefficients:  $\mathbf{u}_j = u_j^n$ . As a result, the Petrov–Galerkin approximation of Equations (13) and (17) is the solution of the following linear algebra problem:

$$\mathbf{K}\mathbf{u} = \mathbf{f}. \quad (26)$$

An important class of Petrov–Galerkin discretizations is obtained from Equation (21) with the choice of test functions  $v = \psi_i = \delta(\mathbf{x}_i)$ , Dirac deltas centered at  $n$  points  $\{\mathbf{x}_i\}_{i=1}^n \subset \Omega$ , where the space  $V^n$  is at least twice differentiable.

In this case, the weak form expressed in equation (11) contains the surface gradient of Dirac deltas, which can only be interpreted in the usual weak sense, by integrating back by parts, and evaluating the result (corresponding to the original strong form of the problem) at the point  $\mathbf{x}_i$ .

The resulting projection method is called a *Collocation Method*, that strongly enforces the original differential equation at the collocation points  $\{\mathbf{x}_i\}_{i=1}^n$ . The linear systems arising from Collocation Methods have the same form of the linear system of Equation (26), but with the following stiffness matrices:

$$\mathbf{K}_{ij} = -\Delta_B \phi_j(\mathbf{x}_i), \quad \mathbf{K}_{ij} = -\Delta_B \phi_j(\mathbf{x}_i) - \phi_j(\mathbf{x}_i) + u_n^2(\mathbf{x}_i)\phi_j(\mathbf{x}_i), \quad (27)$$

respectively for the linear and nonlinear case. In both cases the right hand side is just the evaluation of the forcing term at the collocation points:

$$\mathbf{f}_i = f(\mathbf{x}_i). \quad (28)$$

If the trial and test space coincide,  $V^n = W^n$ , the resulting projection is called a *Galerkin projection*. When necessary, we will use the subscript  $c$  to indicate *collocation* matrices and the subscript  $g$  to indicate *Galerkin* matrices.

### 3. Spectral Methods

In this section, we introduce several instances of the Spectral Element Method, whose common feature is that of describing the computational domain exactly, through the surface NURBS representation coming e.g. from a Computer-Aided-Design (CAD) software. Two members of the family considered in the following coincide with the standard Isogeometric Analysis method and with an instance of the Isogeometric Collocation method. However, we regard that considering them as members of a large family of Spectral Element Methods is profitable in terms of thought and presentation economy.

#### 3.1. Surface representation

The most popular surface representation in the CAD community is that of Non-Rational Uniform B-splines (NURBS). Referring to [36] for an introduction and thorough treatment on this subject, we briefly recall some important definitions. Here we focus only on tensor-product NURBS surfaces, which are a subclass of the much more general family of NURBS parametrizations. Subdivision surfaces [14] or Powell–Sabin B-splines [35, 42] are two possible strategies when non-tensor product surfaces are required.

A NURBS description of a tensor-product bi-variate surface consists of two nondecreasing knot vectors,  $\Theta^i = \{k^1 = 0, k^2, \dots, k^{n_i+p_i}, k^{n_i+p_i+1} = 1\}$ , two positive weight vectors  $W^i = \{w^1, \dots, w^{n_i}\}$  for  $i = 1, 2$ , and a set of  $n = n_1 n_2$  points  $\{\mathbf{P}^{i,j}\}_{i,j=1}^{n_1, n_2} \subset \mathbb{R}^3$ , with  $n_i \geq p_i + 1$ . Here it is assumed that the knot vectors  $\Theta^i$  are *open knot* vectors, meaning that the first and last knots are repeated  $p_i + 1$  times:

$$k^1 = k^2 = \dots = k^{p_i+1} = 0 \quad \text{and} \quad k^{n_i+1} = k^{n_i+2} = \dots = k^{n_i+p_i+1} = 1.$$

The first step in the NURBS parametrization consists in defining the  $n_d$  B-splines of degree  $p$  with respect to the knot vector  $\Theta^d$ ,  $\{B^{(i,p)}\}_{i=1}^{n_d}$ . The B-splines of degree zero are defined by:

$$B^{(i,0)}(s) = \begin{cases} 1 & \text{if } k^i \leq s < k^{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

and then the B-splines of degree  $p$  are constructed by recursion using a convex combination of the B-splines of the previous degree, as described by the Cox–de Boor formula:

$$B^{(i,p)}(s) := \frac{s - k^i}{k^{i+p} - k^i} B^{(i,p-1)}(s) + \frac{k^{i+p+1} - s}{k^{i+p+1} - k^{i+1}} B^{(i+1,p-1)}(s). \quad (30)$$

The one-dimensional NURBS basis functions are defined by:

$$N^{(i,p)}(s) = \frac{w^i B^{(i,p)}(s)}{\sum_{j=1}^{n_d} w^j B^{(j,p)}(s)}, \quad (31)$$

with  $p = p_1, p_2$  depending on  $d = 1, 2$ . The same procedure is repeated for the construction of  $n_2$  splines and NURBS basis functions of degree  $p_2$  in the other direction.

Finally, the surface is parametrized by weighting the grid points' coordinates with a tensor product of NURBS functions:

$$\mathbf{x}(s) := \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{P}^{i,j} N^{(i,p_1)}(s^1) N^{(j,p_2)}(s^2). \quad (32)$$

### 3.2. Approximation spaces

For practical application of the Petrov–Galerkin projection, it is necessary to explicitly construct a basis set for both the approximation spaces  $V^n$  and  $W^n$ . Here we consider only functions on the surface that are tensor product of 1D functions  $\hat{\phi}_i(s)$  defined on the interval  $[0, 1]$ , composed with the inverse surface parametrization  $\mathbf{x}^{-1}$ .

Consequently, the basis functions have the following shape:

$$\phi_i(\mathbf{x}(s)) := \hat{\phi}_{i_1}(s^1) \hat{\phi}_{i_2}(s^2) \quad \mathbf{s} \in \hat{\Omega}, \quad (33)$$



where  $\mathbf{i} = (i_1, i_2)$  is an element of the index set:

$$\mathcal{I} := \{\mathbf{j} = (j_1, j_2), 1 \leq j_1 \leq n_1, 1 \leq j_2 \leq n_2\}. \quad (34)$$

In this work, we consider basis functions on the reference interval  $[0, 1]$ , that originate different flavours of Petrov–Galerkin methods. The common features of the methods presented here are the following:

- in all cases, the map  $\mathbf{x}$  between the reference element and the surface consists of a tensor product of NURBS functions;
- the basis functions are suitable in the sense of [23] for a high-order method, meaning that the interpolation error for a smooth function converges to zero at an exponential rate as the degree of the basis functions is increased.

Two interesting choices for  $\hat{\phi}_i$  are B-spline and NURBS functions as defined in Equations (30) and (31) respectively. Other two possibilities consist in Lagrange interpolants at Gauss–Lobatto points. Here we consider Gauss–Lobatto–Legendre (GLL) points, which are the  $n - 2$  zeros of the derivative of the Legendre Polynomial of degree  $n - 1$ , plus the interval endpoints:

$$s_j^{\text{GLL}} = \begin{cases} j\text{-th zero of } P'_{n-1}(s) & \text{for } j = 1, \dots, n-2 \\ 0 & \text{for } j = n-1 \\ 1 & \text{for } j = n \end{cases}. \quad (35)$$

and Gauss–Lobatto–Chebyshev (GLC) points:

$$s_j^{\text{GLC}} = \cos\left(\pi + \frac{j-1}{n-1}\pi\right) \quad \text{for } j = 1, \dots, n. \quad (36)$$

Then, the Lagrange interpolants are defined as:

$$L_j(s) = \prod_{\substack{i=1 \\ i \neq j}}^n \frac{s - s_i^{\text{GLL}}}{s_j^{\text{GLL}} - s_i^{\text{GLL}}} \quad \text{for } j = 1, \dots, n, \quad (37)$$

for the Lagrange case, and

$$T_j(s) = \prod_{\substack{i=1 \\ i \neq j}}^n \frac{s - s_i^{\text{GLC}}}{s_j^{\text{GLC}} - s_i^{\text{GLC}}} \quad \text{for } j = 1, \dots, n, \quad (38)$$

for the Chebyshev case. In the following, we denote with  $L_j^n$  the  $j$ -th Lagrange interpolant on  $n$  GLL points, and with  $T_j^n$  the  $j$ -th Lagrange interpolant on  $n$  GLC points.

By taking different combinations of basis functions for the trial  $V^n$  and test  $W^n$  spaces, we generate the family of methods listed in Table 1. The names of the different methods are chosen to be self-explanatory and compatible with the literature. For the cases where the trial and test space coincide, the method is

Method name	Acronym	$V^n$	$W^n$
B-spline Galerkin	SG	$B^{(i,p)}$	$B^{(j,p)}$
B-spline Collocation	SC	$B^{(i,p)}$	$\delta(\mathbf{x}(s_j^{\text{Greville}}))$
Isogeometric Galerkin	IG	$N^i$	$N^j$
Isogeometric Collocation	IC	$N^i$	$\delta(\mathbf{x}(s_j^{\text{Greville}}))$
Chebyshev Collocation	CC	$T_i^n$	$\delta(\mathbf{x}(s_j^{\text{GLC}}))$
Chebyshev Galerkin	CG	$T_i^n$	$T_j^n$
Legendre Galerkin	LG	$L_i^n$	$L_j^n$

Table 1: Catalogue of the 7 candidates for high-order methods considered in this work.

classified as a Galerkin method, and any time the test space is formally spanned by Dirac deltas, the method is classified as a Collocation method.

Below we briefly describe each numerical method that will be considered in the following. Our description is by no means exhaustive, and for a treatment of the first four methods of Table 1 we refer to [15], [39] and [2]. However, we let the reader beware that none of these references is concerned with the application of Isogeometric Methods in the context of high-order refinement. For a treatment of high-order methods similar, but not equivalent, to the last three rows of Table 1, we refer to [23], [12], [13].

*B-spline Galerkin Method (SG)*, where the B-spline basis functions defined in (30) are used as basis both for the trial and test spaces.

*B-spline Collocation Method (SC)*. In this case, the B-spline basis functions of equation (30) are used as basis for the trial space, and Dirac deltas centered on the Greville abscissae of the B-spline functions are used as basis for the test space.

*Isogeometric Galerkin Method (IG)*. In this method, the NURBS basis functions defined in (31) are used as basis both for the trial and test spaces. A more consistent name for this method would be *NURBS Galerkin Method*, since however it has been named *Isogeometric Analysis* by its ideators in [26], we stick to this by now standard naming.

*Isogeometric Collocation Method (IC)*. In this case, the NURBS basis functions (31) are used for constructing the trial space, and Dirac deltas centered on the Greville abscissae of the NURBS basis functions are used as basis for the test space.

*Chebyshev Collocation Method (CC)*. For this method, the reference basis functions for the trial space are Lagrange interpolants on Gauss–Lobatto–Chebyshev nodes, and the test functions are Dirac deltas centered in the same Gauss–Lobatto–Chebyshev nodes. The basis functions are constrained to be continuous together with their normal derivative across elements, as done in Chebyshev multipatch methods, see [13, p. 339] for details.

*Chebyshev Spectral Element Method (CG)*. In this method, the basis functions for both trial and test spaces are Lagrange interpolants on Gauss–Lobatto–Chebyshev nodes. The Galerkin method in this case is based on the weak

form (11), with inner products weighted by the function

$$w(x) = \frac{1}{\sqrt{1-x^2}}. \quad (39)$$

As in the previous case, inter-element continuity of the basis functions and their normal derivative is explicitly enforced. One important characteristic of this and the following methods is that the quadrature formulas used for the evaluation of the integrals are based on the same nodes used to define the basis functions. This results in a “variational crime” due to under-integration, but also in a diagonal mass matrix, and allows for efficient integral evaluation. In some books, the methods with integration by quadratures are called “SEM with numerical integration”, or “SEM-NI”.

*Legendre Spectral Element Method (LG).* In this method, the basis functions for both trial and test spaces are Lagrange interpolants on Gauss–Lobatto–Legendre nodes. The basis functions are only continuous across contiguous elements.

To provide some intuition over the different basis functions involved, we draw in Figure 1 the one-dimensional basis functions of degree 7 on the reference interval  $[0, 1]$  for the cases with 2 elements. The B-spline functions are visually similar to the NURBS in many contexts. In Figure 1 the two elements are the intervals  $[0, 0.5]$  and  $[0.5, 1]$ , and in Figure 1 (b) the  $\mathcal{C}^1$  inter-element continuity of the NURBS functions is evident.

### 3.3. Discrete formulation

Although the matrices that appear in the linear Equation (26) could in principle be computed naively as by their definition in Equations (23) and (24), this often is not the most efficient way, especially for high-order methods. In this section we give some details on how the discrete formulations of Section 3.2 could be set up and solved in a reasonably efficient manner.

Significant efficiency savings can be achieved by exploiting the tensor product form of the surface parametrization (Equation (32)) and of the basis function definition (Equation (33)). These tensor product structures allow in turn to express the discrete operators in tensor-product form, that as we shall see drastically reduces the memory and the number of operations required to compute a matrix-vector product. This fact holds true both for methods based on the B-spline family and for methods based on Lagrange interpolants. For a detailed reference on the efficient implementation of tensor-product high-order methods, we refer to [20].

The first step in the discretization of the variational form (13) is the evaluation of the inner products  $(\nabla u, \nabla v)$  on all the basis functions. This operation can be performed by pulling back the integrals to the reference domain  $\widehat{\Omega} := [0, 1]^2$ , as

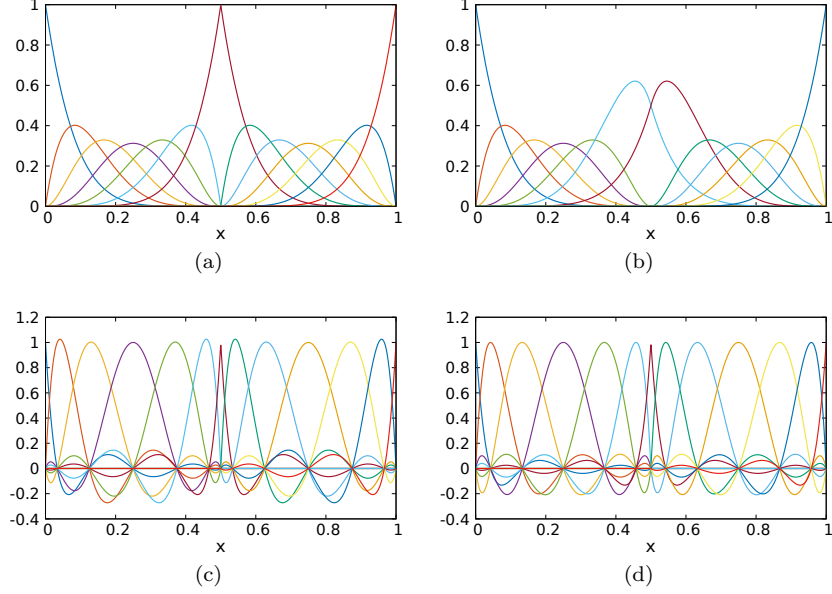


Figure 1: Basis functions of degree 7 on the reference interval for the case of two equal elements for  $C^0$  NURBS (a),  $C^1$  NURBS (b), Chebyshev Spectral Element (c), Legendre Spectral Element (d) methods.

follows:

$$\begin{aligned}
 \int_{\Omega} \nabla \phi_i(\mathbf{x}) \cdot \nabla \phi_j(\mathbf{x}) d\mathbf{x} &= \sum_{k=1}^3 \int_{\Omega} \frac{\partial \phi_i(\mathbf{x})}{\partial x^k} \frac{\partial \phi_j(\mathbf{x})}{\partial x^k} d\mathbf{x} \\
 &= \sum_{k=1}^3 \sum_{\alpha, \beta=1}^2 \int_{\hat{\Omega}} \frac{\partial \hat{\phi}_i(\mathbf{x}(\mathbf{s}))}{\partial x^k} \frac{\partial x^k}{\partial s_{\alpha}} \frac{\partial \hat{\phi}_j(\mathbf{x}(\mathbf{s}))}{\partial x^k} \frac{\partial x^k}{\partial s_{\beta}} J(\mathbf{s}) d\mathbf{s} =: K_{ij},
 \end{aligned} \tag{40}$$

where  $J = \sqrt{|\det g_{\alpha\beta}|}$ , and the hat is used to denote quantities defined on the reference domain. The Neumann boundary terms are also evaluated on the boundary of the reference domain,  $\partial \hat{\Omega}_N$ :

$$\begin{aligned}
 \int_{\partial \Omega_N} \phi_i(\mathbf{x}) \nabla \phi_j(\mathbf{x}) \cdot \boldsymbol{\nu} d\mathbf{x} &= \int_{\partial \Omega_N} \phi_i(\mathbf{x}) \sum_{k=1}^3 \frac{\partial \phi_j(\mathbf{x})}{\partial x^k} \nu^k d\mathbf{x} \\
 &= \int_{\partial \hat{\Omega}_N} \hat{\phi}_i(\mathbf{s}) h_N(\mathbf{s}) J(\mathbf{s}) d\mathbf{s},
 \end{aligned} \tag{41}$$

where  $\hat{\boldsymbol{\nu}} = (\hat{\nu}^1, \hat{\nu}^2)$  is the outer normal to the reference domain  $\hat{\Omega}$ , and  $\hat{\nu}$  is given in components by:

$$\nu^k = \hat{\nu}^{\alpha} \frac{\partial x^k}{\partial s^{\alpha}}. \tag{42}$$

The integrals on the right hand side of Equations (40) and (41) are then evaluated by quadratures. To this end, let  $\{(\xi_k, \omega_k)\}_{k=1}^{Q_1}$  be the quadrature point-quadrature weight couples for the first coordinate, and  $\{(\eta_l, \varrho_l)\}_{l=1}^{Q_2}$  the quadrature points and weights for the second coordinate. Here  $Q_1$  and  $Q_2$  denote the number of quadrature points chosen for the two directions. It is convenient to introduce the auxiliary matrices:

$$(G_g^{\alpha\beta})_{kl} = g^{\alpha\beta}(\xi_k, \eta_l) J(\xi_k, \eta_l) \omega_k \varrho_l \quad (43)$$

and

$$(M)_{ki} = \widehat{\phi}_i(\xi_k) \quad (D_1)_{ki} = \widehat{\phi}'_i(\xi_k). \quad (44)$$

The right hand side of equation (40) can be approximated by:

$$K_g = \begin{pmatrix} M \otimes D_1 \\ D_1 \otimes M \end{pmatrix}^T \begin{bmatrix} G_g^{11} & G_g^{12} \\ G_g^{21} & G_g^{22} \end{bmatrix} \begin{pmatrix} M \otimes D_1 \\ D_1 \otimes M \end{pmatrix} \quad (45)$$

which is a sequence of matrix-matrix products written in block tensor form. The special block-tensor product structure of the expression (45) allows for a reduction of the computational cost with respect to a general basis lacking the tensor product structure. Indeed, suppose that in both directions there are  $N$  basis functions, and that the same  $Q$ -point quadrature rules are applied. Then, the number of operations required for the evaluation of the integral (40) decreases from  $O(Q^2 N^2)$  to  $O(Q^2 + 2QN)$ .

The Neumann boundary term in Equation (41) is computed similarly, and it represents a known term in the resulting algebraic system.

The action of the Laplace–Beltrami operator in the collocative case is even simpler, since it is sufficient to evaluate the derivatives of the basis functions and of the metric tensor on the collocation points, and then assemble point by point the coordinate expression of Equation (8). More precisely, let us introduce the collocation matrix for the second order derivatives of the basis functions:

$$(D_2)_{ki} = \widehat{\phi}''_i(\xi_k), \quad (46)$$

we define the matrices  $G_c^{\alpha\beta}$  for the collocation case as:

$$(G_c^{\alpha\beta})_{kl} = g^{\alpha\beta}(\xi_k, \eta_l), \quad (47)$$

and we introduce the pointwise evaluation of the Christoffel symbols:

$$(C^\mu_{\alpha\beta})_{kl} = g^{\alpha\beta}(\xi_k, \eta_l) \Gamma^\mu_{\alpha\beta}(\xi_k, \eta_l). \quad (48)$$

The resulting discretization of the Laplace–Beltrami operator is:

$$K_c = \begin{bmatrix} G_c^{11} & G_c^{12} \\ G_c^{21} & G_c^{22} \end{bmatrix} \begin{bmatrix} M \otimes D_2 & D_1 \otimes D_1 \\ D_1 \otimes D_1 & D_2 \otimes M \end{bmatrix} + \begin{bmatrix} C_{11}^1 & C_{21}^2 \\ C_{21}^1 & C_{21}^2 \end{bmatrix} \begin{pmatrix} M \otimes D_1 \\ D_1 \otimes M \end{pmatrix} + \begin{bmatrix} C_{12}^1 & C_{12}^2 \\ C_{22}^1 & C_{22}^2 \end{bmatrix} \begin{pmatrix} M \otimes D_1 \\ D_1 \otimes M \end{pmatrix}. \quad (49)$$

We remark that in Equation (49), the collocation matrices  $G_c^{\alpha\beta}$ , defined according to Equation (47), are different from the Galerkin matrices  $G_g^{\alpha\beta}$ , defined according to Equation (43),

For the Chebyshev-SEM method, the stiffness matrix is computed exactly as in Equation (49), replacing  $G_c^{\alpha\beta}$  with  $G_g^{\alpha\beta}$  and replacing  $C_{\alpha\beta}^\mu$  with:

$$(S_{\alpha\beta}^\mu)_{kl} = g^{\alpha\beta}(\xi_k, \eta_l) \Gamma_{\alpha\beta}^\mu(\xi_k, \eta_l) J(\xi_k, \eta_l) \omega_k \varrho_l. \quad (50)$$

For both Galerkin and Collocation methods based on Chebyshev points, Neumann boundary conditions can be imposed by collocation. This can be achieved by replacing the rows related to the collocation points lying on  $\partial\Omega_N$  with:

$$\sum_l \phi_i(\xi_N, \eta_l) \nabla \hat{\phi}_j(\xi_N, \eta_l) \cdot \hat{\nu}(\xi_N, \eta_l), \quad (51)$$

where to simplify the presentation we made the hypothesis that the Neumann boundary is located at the points with  $s^1 = \xi_N$  for all  $s^2$ , and the corresponding rows on the right hand side should be replaced by the known value  $h_N(\xi_N, \eta_l)$ .

For B-spline and Isogeometric methods, the imposition of Neumann boundary conditions by collocation is still an active research area. Since there are no simple, established solution to this problem, we do not discuss it here, and refer instead to [4, 34].

Computing the system's matrix in the Allen–Cahn case requires one more step, namely the efficient evaluation of the integral coming from the fixed point linearization

$$\int_{\Omega} \phi_i(\mathbf{x}) u_n^2(\mathbf{x}) \phi_j(\mathbf{x}) d\mathbf{x} = \int_{\hat{\Omega}} \hat{\phi}_i(\mathbf{s}) u_n^2(\mathbf{x}(\mathbf{s})) \hat{\phi}_j(\mathbf{s}) J(\mathbf{s}) d\mathbf{s}. \quad (52)$$

This additional term is discretized as:

$$(\psi_{i(i,j)}, u_n^2 \phi_{j(k,l)}) = (M_{ri} \otimes M_{sj}) N_{rs} (M_{rk} \otimes M_{sl}), \quad (53)$$

where the index sets  $i(i,j)$  and  $j(k,l)$  are introduced to pass from the two-dimensional matrix notation on the left hand side to the multi-dimensional indexing of the right hand side. On the right hand side the summation with respect to  $r$  and  $s$  is implied. The nonlinearity  $N$  can either be evaluated at the quadrature points as:

$$N_{rs} = (\phi_{\mathbf{k}}(\xi_r, \eta_s) u_{n,\mathbf{k}})^2 J(\xi_r, \eta_s) \omega_r \varrho_s \quad (54)$$

in the case of Galerkin methods, or at the collocation points as:

$$N_{rs} = (\phi_{\mathbf{k}}(\xi_r, \eta_s) u_{n,\mathbf{k}})^2, \quad (55)$$

for collocation methods. In Equations (54) and (55), the symbol  $\phi_{\mathbf{k}} u_{n,\mathbf{k}}$  is a shortcut for the sum:

$$\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \phi_i \phi_j u_{n,ij}. \quad (56)$$

Note that in Equation (56), the index  $n$  refers to the  $n$ -th fixed point iteration, while  $i$  and  $j$  sum over all the basis functions.

In the case of Galerkin methods, aliasing errors, if present, may be reduced by evaluating  $N$  via higher-order quadrature rules. Aliasing, however, has not been an issue in the present work, since the resolution was sufficient to represent the nonlinearity with a good precision. For details on how aliasing may affect a computation not sufficiently resolved, see [10].

In the case of multi-element or multi-patch discretizations, the steps outlined above are repeated elementwise, and summed in the global stiffness matrix with the appropriate numbering of the degrees of freedom, as in standard Finite Element codes.

### 3.4. Essential boundary conditions

After treating natural boundary conditions in Section 3.3, we now discuss essential, or Dirichlet, boundary conditions. For the methods based on Lagrange interpolants, the essential boundary conditions are imposed by row elimination. The rows related to the boundary degrees of freedom are replaced by the corresponding rows of the identity matrix:

$$K_{ij} = \delta_{ij} \quad \forall i \text{ such that } \mathbf{x}_i \in \partial_D \Omega, \quad (57)$$

and at the right hand side, the Dirichlet datum is imposed at the corresponding degree of freedom:

$$\mathbf{f}_i = h_D(\mathbf{x}_i) \quad \forall i \text{ such that } \mathbf{x}_i \in \partial_D \Omega. \quad (58)$$

For the methods based on B-spline or NURBS basis functions, such a direct approach is not feasible since the basis is not interpolatory. We resort to a least-squares enforcement of essential boundary conditions. The least-squares problem requires a set of points on the boundary  $\{\bar{\mathbf{x}}_i\}_{i=1}^q$  and an index-set  $\mathbf{i}$  where the indices of the basis functions different from zero at the boundary are stored. The basis functions indexed by  $\mathbf{i}$  are evaluated at the boundary points, forming the matrix  $V \in \mathbb{R}^{q \times n_b}$ :

$$V_{ij} = \phi_{\mathbf{i}(j)}(\bar{\mathbf{x}}_i) \quad \text{for } j = 1, \dots, n_b, i = 1, \dots, q \quad (59)$$

where  $n_b$  is the cardinality of  $\mathbf{i}$ . The least-squares problem is well-posed if and only if  $q \geq n_b$ . Similarly, we introduce the array  $\mathbf{q} \in \mathbb{R}^q$ , containing the evaluation of the Dirichlet datum on the interpolation points:

$$\mathbf{q}_j = h_D(\bar{\mathbf{x}}_j) \quad \text{for } j = 1, \dots, q, \quad (60)$$

and the boundary restriction matrix  $Q \in \mathbb{R}^{n_b \times n}$ . We also introduce the following spaces:

$$V_o^n = \{v \in V^n : \gamma_{\partial\Omega} v = 0\}, \quad (61)$$

$$V_o^n = V^n \setminus V_o^n = \text{span}\{\phi_{\mathbf{i}(j)}\}_{j=1}^{n_b}, \quad (62)$$

and we denote by  $\mathbf{y}_\circ$  any vector of  $\mathbb{R}^n$  containing the expansion coefficients in  $\{\phi_j\}$  of any function with vanishing trace on the Dirichlet part of the boundary:

$$(\mathbf{y}_\circ)_{\mathbf{i}(j)} = 0 \quad \text{for } j = 1, \dots, n_b, \quad (63)$$

and with  $\mathbf{y}_\partial$  we denote any vector of  $\mathbb{R}^n$  containing the expansion coefficients of any function whose trace is not identically zero on  $\partial_D \Omega$ , and whose interior coefficients vanish:

$$(\mathbf{y}_\partial)_j = 0 \quad \text{for } j \notin \{\mathbf{i}(k)\}_{k=1}^{n_b}. \quad (64)$$

Notice that functions in  $V_\partial^n$  are in general not vanishing in the interior of the domain pointwise. They decay as we approach the interior of the domain in a mesh dependent way, following the decay of the B-spline basis functions whose value on the boundary is non-zero.

The restriction operator is then defined as:

$$\begin{cases} Q\mathbf{y}_\circ = \mathbf{0} \\ QQ^T\mathbf{y}_\partial = \mathbf{y}_\partial \end{cases} \quad (65)$$

We remark that  $Q^T : \mathbb{R}^{n_b} \rightarrow \mathbb{R}^n$  takes a vector of boundary “values” and extends it to zero on the interior, and  $Q^T Q \in \mathbb{R}^{n \times n}$  returns the boundary lifting of a vector.

The least squares imposition of Dirichlet boundary conditions is achieved through the augmented linear system:

$$\begin{bmatrix} K & Q^T V^T V \\ V^T V Q & 0 \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ V^T \mathbf{q} \end{pmatrix}, \quad (66)$$

where  $\lambda \in \mathbb{R}^{n_b}$  is a vector of auxiliary Lagrange multipliers. A side-effect of this augmentation is the growth of the matrix condition number, but in our numerical experiments this is contained to a factor of  $\simeq 100$ . The number of unknowns usually does not increase by more than  $\simeq 20\%$ , but this depends on the boundary to surface ratio of the domain. Alternatively, one could impose the boundary conditions by Nitsche method, as done, for example, in [38].

### 3.5. Cost estimates

The different choices of test and trial functions have an impact also on the computational cost of each method. For high order methods, it is convenient to avoid assembling the full system matrix, and to compute only its action on a vector. This can be achieved by a sequence of matrix-matrix products involving local element matrices. Calling  $Q$  and  $n$  the number of quadrature points and the number of basis functions in each direction for each element, we have the following cost entries [20]:

- $28Q^2n^4$  for the matrix-matrix multiplications of Equation (45), for the case of B-spline and NURBS Galerkin Methods;



Table 2: Leading-order cost for an elementwise matrix-matrix multiplication, for the 7 numerical methods of Section 3.2.

SG	SC	IG	IC	CC	CG	LG
$112n^6$	$28n^4$	$112n^6$	$28n^4$	$16n^4$	$20n^6$	$20n^6$

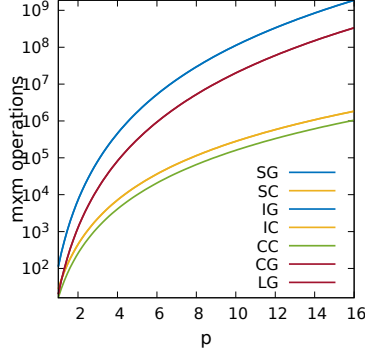


Figure 2: Operation count for an elementwise matrix-matrix multiplication, for the 7 numerical methods of Section 3.2. Note that some of the methods share the same color, since the operation count is the same and the relative curves overlap. More precisely, the SG and IG are both shown in blue, SC and IC are both in yellow, CG and LG are both red.

- $20Q^2n^4$  for the matrix-matrix multiplications of Equation (45), for the Legendre and Chebyshev Galerkin Methods;
- $28n^4$  for the matrix-matrix multiplications of Equation (49), for the B-spline and NURBS Collocation Methods;
- $16n^4$  for the matrix-matrix multiplications of Equation (49), for the Chebyshev Collocation Method.

In most practical codes, the matrix-matrix multiplications are the subroutines absorbing most of the computational resources and time. For this reason, we proceed discussing the computational complexity of this performance-critical phase.

Following the common practice, in the quadrature rules we take  $Q = n$  for the Legendre and Chebyshev methods, and  $Q = 2n - 1$  for the B-spline and NURBS methods.

As a result, for a fixed polynomial order  $p$ , we have a leading-order operation count for the matrix-matrix multiplication that differs quite significantly for the different numerical methods, that we summarize in Table 2. The same information is visualized in Figure 2.

While this choice guarantees exact polynomial integration, there is a vast literature dedicated to better choices of quadrature formulas for Isogeometric

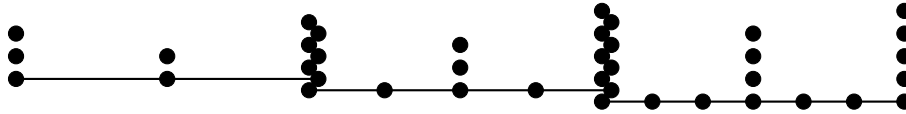


Figure 3: Evolution of a knot vector under  $k$ -refinement. Each dot represents a knot, with the convention that a vertical array of  $m$  dots stands for a knot of multiplicity  $m$ . The initial knot vector (left) is given. The knot vector in the middle is obtained after  $k$ -refinement of the initial knot vector. The knot vector on the right is obtained after two levels of  $k$ -refinement.

methods that shows how the computational cost argument could be less stringent than the one depicted here (see for example [27, 3, 40]). We point out that the coefficients in Table 2 could be smaller for the B-spline and Isogeometric methods, see in particular the new memory-efficient assembly strategies introduced in [11].

#### 4. Numerical results

In this section we compare the numerical methods presented in Section 3 for the Laplace–Beltrami and Allen–Cahn equations on moderately complex surfaces. The numerical experiments are designed to assess the behaviour of the different methods with respect to order elevation. Due to the high flexibility in the definition of B-spline and NURBS basis functions, there are many ways to construct and refine such basis. Two common ways to increase the order of B-spline and NURBS functions are  $p$  and  $k$ -refinement (see e.g. [15], [26], [8]). While for  $p$ -refinement there is a definition the literature agrees on, namely the increase by one of all the knots’ multiplicity, a single step of  $k$ -refinement may consist of a  $p$ -refinement followed by some (somewhat arbitrary) knot insertions, at locations that do not coincide with existing knots. The definition of a single  $k$ -refinement step adopted here consists in increasing by one the multiplicity of both the internal and the end knots, followed by the insertion of one internal knot per knot interval, with multiplicity equal to one. Subsequent  $k$ -refinements do not start from previous stages, but from the knot vector of the original geometry, i.e., the internally inserted knots are removed before elevating the degree of the B-splines.

In particular, given an initial knot vector defining the geometry, in this work we construct the new knot vector obtained after  $m$  steps of  $k$ -refinements by taking the same knot vector, with the multiplicities of every knot increased  $m$  times, union with a vector of  $m$  new knots of multiplicity one for each knot interval, located in new points equally spaced between each couple of subsequent knots in the original vector. An example that shows our convention is available in Figure 3.

We remark that  $k$ -refinement can be interpreted as a composition of a  $p$ -refinement obtained by augmenting the geometry knots’ multiplicity, and a series of knot insertions, obtained by inserting new inner knots with multiplicity one.

Both  $p$  and  $k$ -refinement are a form of degree elevation, with the important difference that during  $p$ -refinement the basis functions keep their original global

continuity, while after  $m$  steps of  $k$ -refinement, new internal knots are inserted in the knot vector, generating basis functions at the inserted knots that have the maximum available continuity. For example, if initially the basis functions have order  $q$  and are globally  $\mathcal{C}^s$ , after  $m$  steps of  $p$ -refinement the new basis functions will have order  $q + m$ , and maintain the global regularity  $\mathcal{C}^s$ . The same initial basis functions, after  $m$  steps of  $k$ -refinement will have order  $q + m$ , global regularity  $\mathcal{C}^s$ , and local regularity  $\mathcal{C}^{q+m-1}$  on the newly inserted knots.

An alternative approach to the  $k$ -refinement strategy described above, consists in removing internal knots while increasing the degree of the B-splines. This strategy, also referred to as  $k$ -coarsening, implies an increase in the global regularity of the B-spline basis functions, at the price of generating non-nested spaces. This approach has the advantage that the number of degrees of freedom does not grow too fast with the polynomial degree, and may also lead to better conditioned matrices than those obtained by keeping fixed the global continuity of the basis functions. A major disadvantage of this approach is related to the fact that the geometry cannot be preserved through  $k$ -coarsening, making it only useful for trivial geometries, and requiring an additional geometry reconstruction step, which may not be well posed, or may give unsatisfactory results. In this work we only show the  $k$ -refinement strategy illustrated above, which is guaranteed to preserve the exact geometry.

In all cases the linear systems are assembled with the `numpy` [44] and `igakit` [17] libraries.

#### 4.1. Description of the test cases

A comprehensive test requires that the following possibilities are fully considered:

- flat surfaces (domains) and curved surfaces;
- collocation and Galerkin;
- mixed boundary conditions;
- $p$ -refinement and  $k$ -refinement (when applicable);
- linear and nonlinear problems.

In addition, it is sensible to check that for B-spline and NURBS collocation methods, the convergence rate does not depend drastically on the choice of collocation points.

The test cases we set up consist in a homogeneous Laplace–Beltrami problem on two geometries of increasing complexity, including a comparison between different collocation strategies, the analysis of a mixed Neumann–Dirichlet boundary condition problem, and a nonlinear test case solving the Allen–Cahn equation.

In addition to standard  $p$ -refinement strategies, we also make a comparison between hybrid NURBS-SEM methods and a variant of the isogeometric  $k$ -refinement strategy. For consistency, in this case the comparison is not performed

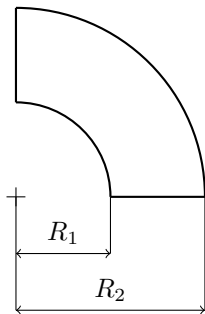


Figure 4: Surface geometry for the test case n.1.

on the basis of the polynomial degree, but on the total number of degrees of freedom, since the  $k$ -refinement strategy that we employ introduces a higher number of degrees of freedom for a fixed polynomial order than standard  $p$ -refinement in SEM.

The first surface we consider consists of a domain of  $\mathbb{R}^2$ , lying on a plane oblique to the three coordinate axis of  $\mathbb{R}^3$ . Since the surface is flat, any harmonic function in the plane of the surface will also solve the Laplace–Beltrami equation. We consider a transcendental harmonic function as a reference solution, whose expression is reported in Table 3. The restriction of this function to the surface boundary provides the required Dirichlet data.

The flat geometry that we consider is a quarter of annulus, shown in Figure 4, obtained by considering the region between two concentric circles with inner radius  $R_1 = 0.5$  and outer radius  $R_2 = 1$ , and two orthogonal diameters.

The annular surface is represented by a mesh of  $2 \times 2$  elements, as shown in Figure 5 (a). The collocation points for the Chebyshev (GLC nodes) and Isogeometric (Greville nodes) methods of degree 7 are shown in Figures 5 (b) and (c) respectively.

The curved geometry considered for this test is shown in Figure 6, and is obtained by revolving a C-shaped profile around a quarter of a circle of diameter equal to 5 times the sectional height. Since this surface has nonzero curvature, an harmonic function in  $\mathbb{R}^n$  will not in general be a solution of the Laplace–Beltrami equation. Consequently, in this case the numerical results are compared against a manufactured solution, as reported in Table 3. For this geometry we construct a mesh with 3 elements in the direction of revolution, and in 5 elements in the radial direction. In Figure 6 we show the collocation points for a Chebyshev collocation method (a) and for a B-spline or NURBS collocation method (b), both of degree 7.

The full set of numerical experiments carried out in the following sections is summarized in Table 3.

#### 4.2. $p$ -refinement, flat geometry

The results for the case with annular geometry and a transcendental harmonic solution are shown in Figure 7. Figure 7 (a) shows that for a polynomial degree

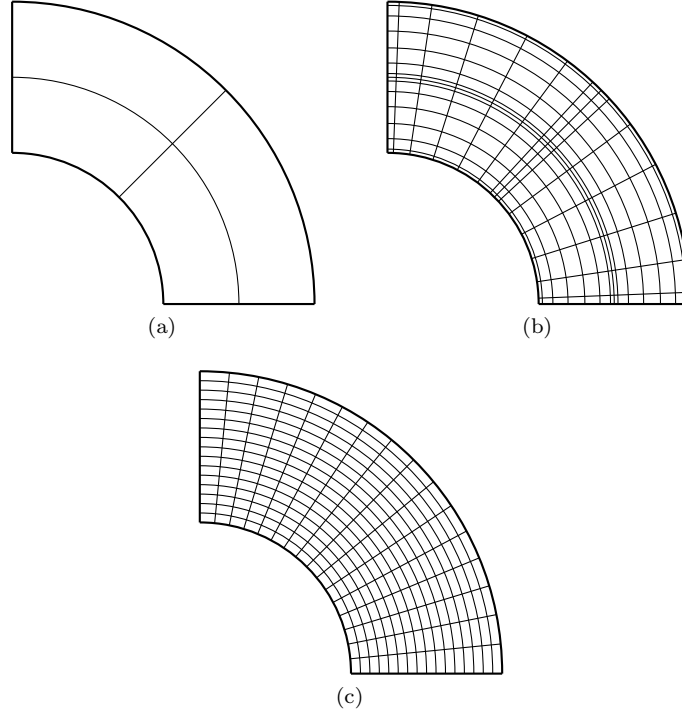


Figure 5: Discretization of the quarter of annulus in four elements (a); Gauss-Lobatto-Chebyshev collocation points (b); Greville collocation points (c).

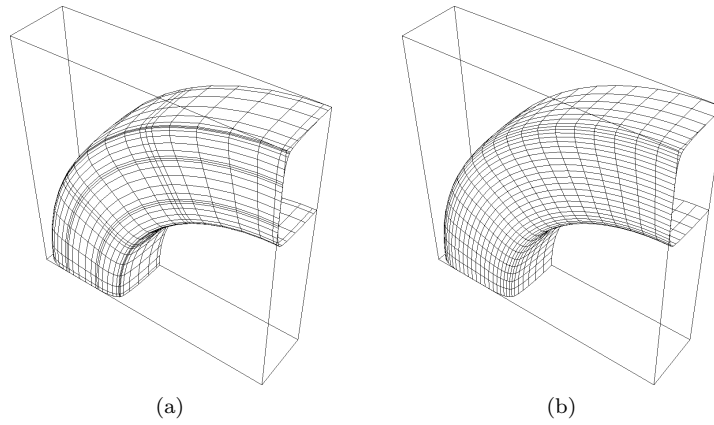


Figure 6: Discretization in  $3 \times 5$  elements of the C-shaped surface for the second test case: Gauss-Lobatto-Chebyshev collocation points (a); Greville collocation points (b).

Table 3: Summary of the numerical experiments performed in this work to compare the high-order methods of Section 3.

$n$	description	geometry	solution $u(\mathbf{x})$
1	$p$ -ref flat	Annulus	$-\frac{1}{2\pi} \log  \mathbf{x} - (1, 1, 0) $
2	collocation iga	Annulus	$-\frac{1}{2\pi} \log  \mathbf{x} - (1, 1, 0) $
3	$p$ -ref mixed bc	Annulus	$-\frac{1}{2\pi} \log  \mathbf{x} - (1, 1, 0) $
4	$p$ -ref curved	C-surface	$\cos(x_2) \cos(x_3)$
5	$k$ -ref flat	Annulus	$-\frac{1}{2\pi} \log  \mathbf{x} - (1, 1, 0) $
6	$k$ -ref nonlinear	Annulus	$x_1^2 - x_2^3$

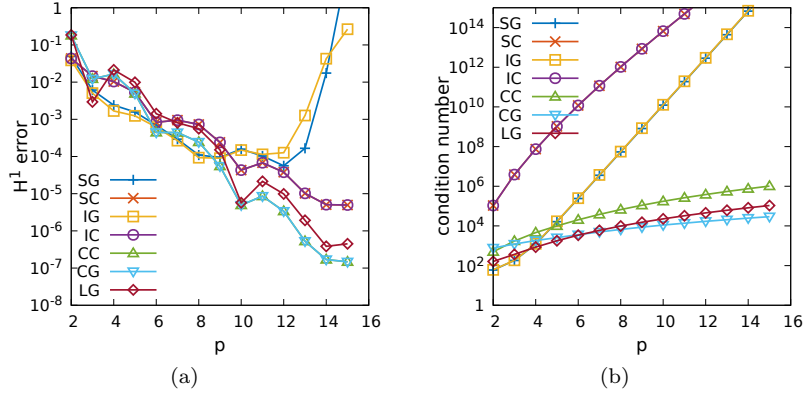


Figure 7: Comparison of the numerical methods for the annular geometry and the transcendental solution  $u(\mathbf{x}) = \log |\mathbf{x} - (1, 1, 0)|$ . Error in the  $H^1$  norm (left) and matrix condition number (right) as function of the polynomial degree.

between 2 and 9 all the methods considered here achieve the expected exponential convergence. For polynomial degrees up to 11, the B-spline and NURBS method except for the IGA-Collocation method fail at keeping the exponential trend, and for even higher polynomial degrees, only the Spectral Element methods show a satisfactory behaviour, reaching spectral accuracy.

The observed behaviour can be explained by looking at the matrix condition number as a function of the polynomial degree  $p$ , shown in Figure 7 (b). From this picture, it is clear that B-spline and NURBS matrices are too ill conditioned for high orders. Conversely, the condition number of the Chebyshev and Legendre matrices increases only algebraically, with the expected asymptotic order  $O(p^4)$ .

#### 4.3. Dependence on the choice of collocation points

The problem set up with annular geometry and transcendental solution is at the basis of two further tests regarding respectively the choice of collocation points in Isogeometric analysis and the use of mixed Neumann–Dirichlet boundary conditions.

To this end, we repeat the test described above for the Isogeometric collocation method with three different sets of collocation points, namely:

**Greville points** Greville points are defined as the points in the unit interval where each basis function achieves its local maximum.

**Demko points** also called *Demko abscissae*, were introduced in [19] and applied in [2] to Isogeometric Collocation. These are the points at which Chebyshev splines (i.e. splines which oscillate between  $-1$  and  $1$ ) achieve a maximum or a minimum. By definition, the maxima and minima of Chebyshev splines are exactly  $1$  and  $-1$  respectively.

**Optimized points** In this case, a minimizer package from the library `scipy` [28] is used to find the collocation points that minimize the  $H^1$  norm of the error for each degree  $p$  of the basis functions. This set of points clearly is out of reach if the solution is not known in advance, or if a sharp error estimator is not available. However, it is instructive to compute anyways this set of points as a lower bound for the comparison with Greville and Demko points.

Only for this test, due to the very high computational cost of the optimization subroutine, we subdivide the geometry in a single large element instead of four smaller elements. The results of this comparison, shown in Figure 8, show that on average Demko points may lead to slightly smaller errors than Greville points, but the convergence rate and the maximum attainable accuracy seem not to be much influenced by the choice of collocation points. However, it is worth mentioning that many other choices of collocation points are available in the literature (see, for example, [1]), and that the list we provide here is by no means complete.

#### 4.4. Problems with mixed boundary conditions

To check the treatment of Neumann boundary conditions, we replace the Dirichlet boundary condition in the curved edge of radius  $R_2 = 1$  with Neumann boundary conditions.

The results of this last test case, shown in Figure 9, confirm the good behaviour of Spectral Element methods with respect to degree elevation. The imposition of natural boundary conditions in Galerkin methods appears to be better conditioned than in collocation methods.

#### 4.5. $p$ -refinement, curved geometry

The results using the manufactured solution given in Table 3 are shown in Figure 10, where we plot on the left the  $H^1$  error and on the right the matrix condition number as a function of the polynomial degree. We report in Figure 11 a plot of the solution (a) together with a map of the pointwise error (b) for the Legendre case of order 15. As for the previous experiments, we see that B-spline and NURBS methods fail to achieve spectral accuracy, while performing very well at relatively low polynomial degrees.

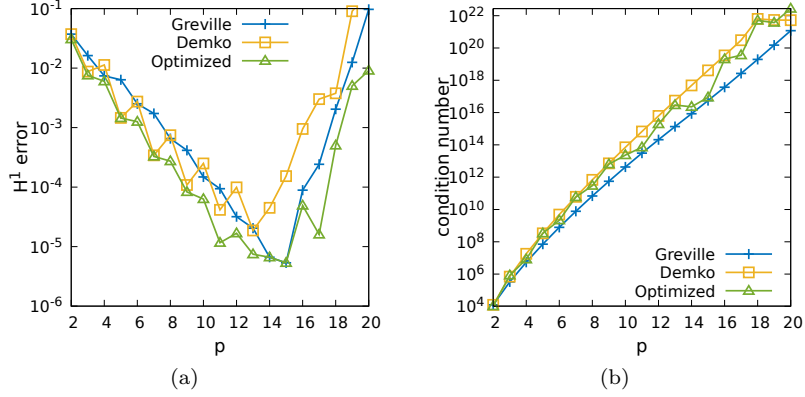


Figure 8: Comparison of the three Isogeometric collocation methods based on three sets of collocation points. Error in the  $H^1$  norm (left) and matrix condition number (right) as function of the polynomial degree.

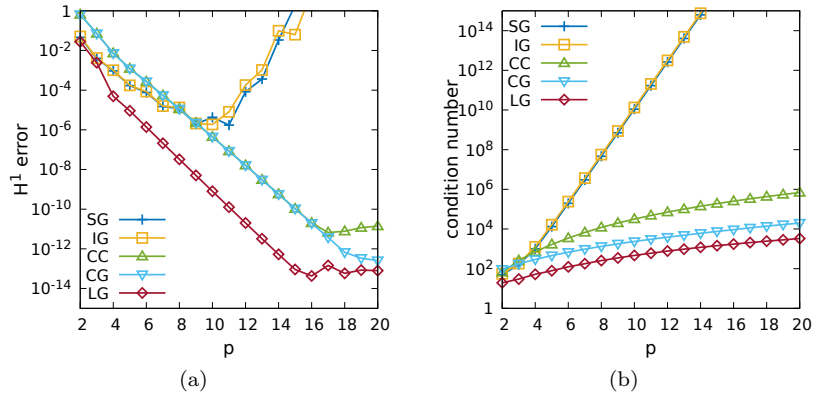


Figure 9: Comparison of the numerical methods for the annular geometry with transcendental solution and Neumann boundary conditions. Error in the  $H^1$  norm (left) and matrix condition number (right) as function of the polynomial degree.



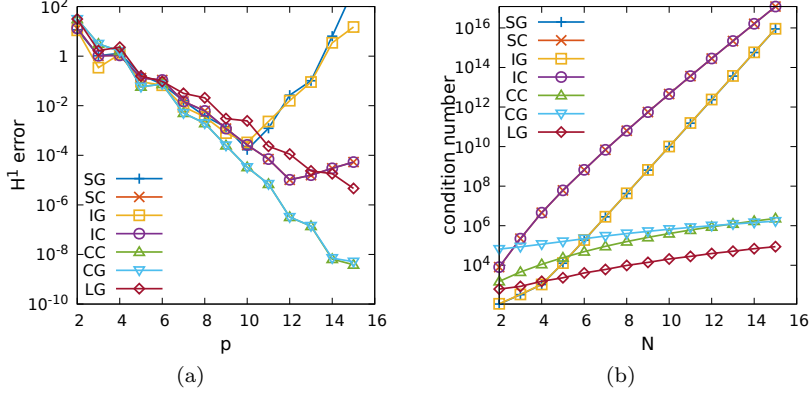


Figure 10: Comparison of the numerical methods for the C-shaped surface and the transcendental solution  $u(\mathbf{x}) = \cos(x_2)\cos(x_3)$ . Error in the  $H^1$  norm (left) and matrix condition number (right) as function of the polynomial degree.

#### 4.6. $k$ -refinement, flat geometry

In view of the results shown in Sections 4.2 and 4.5, it may be tempting to conclude that IGA underperforms in terms of  $p$ -refinement if compared with a hybrid NURBS-mapped Spectral Element Method. However, any comparison of numerical methods focusing on the behaviour with respect to degree elevation of the underlying basis functions would not be complete if  $k$ -refinement were not considered. An important feature of our  $k$ -refinement strategy is that it allows the addition of internal knots while increasing the polynomial degree. This cannot be achieved by simple  $p$ -refinement, and may give an edge to  $k$ -refinement over  $p$ -refinement in terms of accuracy, thanks to these additional degrees of freedom.

The goal of the present section is therefore to compare the performance of increasing the order of the method by performing  $m$  steps of  $k$ -refinement in B-spline and NURBS methods with  $m$  standard steps of  $p$ -refinement in Spectral Element Methods.

The tests are performed on the quarter of annulus geometry, with initial knot vectors given by:

$$\Theta^1 = \{0, 0, 0, 0.5, 0.5, 1, 1, 1\} \quad \Theta^2 = \{0, 0, 0, 1, 1, 1\}. \quad (67)$$

As for the previous cases, we consider a transcendental solution reported in Table 3. At step  $m$ , we generate the new knot vectors by increasing the multiplicity of each of the above knots by  $m$ , elevating the degree of the polynomial base by  $m$ , and inserting  $2m$  knots in  $\Theta^1$ , of which  $m$  equally spaced between  $(0, 0.5)$ ,  $m$  equally spaced between  $(0.5, 1)$ , and finally  $m$  new knots in  $\Theta^2$ , equally spaced between  $(0, 1)$ .

In all the test cases, Isogeometric methods with  $k$ -refinement are remarkably efficient for polynomial degrees up to 10, where on average these deliver a solution

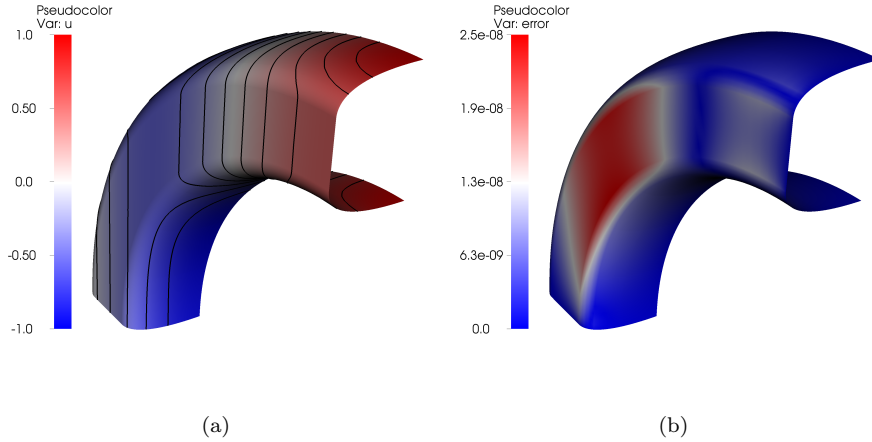


Figure 11: Plot of the solution  $u(\mathbf{x}) = \cos(x_2)\cos(x_3)$  (a) and of the pointwise error for the LG method with polynomial degree equal to 15 (b).

two orders of magnitude more accurate than their SEM counterparts with the same polynomial degree. This is to be expected, since the  $k$ -refinement procedure generates a set of basis functions which is  $2m^2$  bigger than the corresponding  $p$ -refinement SEM basis functions, and makes the comparison between the two methods unfair if done in terms of the polynomial degree alone.

A more fair comparison is obtained when the error and the condition number are plotted in terms of the number of basis functions, as in Figure 12. In this case, the accuracy per degree of freedom is substantially the same, up to order 10. As the polynomial degree is increased, the condition number of the Isogeometric matrices becomes so high (as shown in Figure 14 (a)) that no meaningful solution is delivered already at  $p = 12$ . Conversely, SEM achieved spectral accuracy in all test cases.

Arguably, one could consider a comparison between  $hp$ -refinement on the spectral methods and  $k$ -refinement on the isogeometric methods. It is in principle possible to add  $m$  internal knots before performing  $m$  degree elevation, and to compare this strategy to our  $k$ -refinement strategy (perform  $m$  degree elevation and then add  $m$  internal knots).

In this case, the number of degrees of freedom in the  $hp$ -refinement would grow much more quickly than our  $k$ -refinement strategy, and a comparison on the basis of the number of degrees of freedom would lead to a large imbalance of the polynomial degrees for the same number of degrees of freedom, making this comparison less significant.

#### 4.7. $k$ -refinement, nonlinear problem

The nonlinear test consists in solving the Allen–Cahn equation on the surface shaped as a quarter of annulus, with a forcing term  $f$  chosen so that the exact

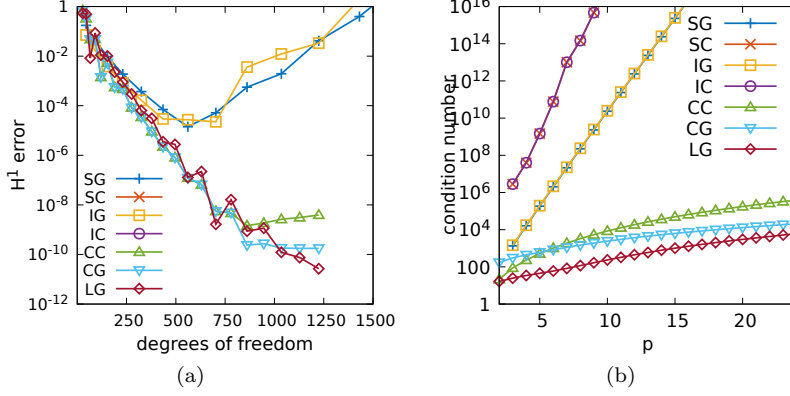


Figure 12: Comparison of the  $k$ -refinement methods for the annular geometry and the transcendental solution  $u(\mathbf{x}) = \log |\mathbf{x} - (1, 1, 0)|$ . Error in the  $H^1$  norm as function of the number of degrees of freedom (left) and matrix condition number as function of the polynomial degree (right).

solution is  $u(\mathbf{x}) = x_1^2 - x_2^3$ . The tolerance for the fixed point methods, computed as defined in Equation (19), is set to  $10^{-15}$ . The results of this test are shown in Figure 13, where  $p$ -refinement for the SEMs is compared with  $k$ -refinement for B-spline and Isogeometric methods. The results confirm what seen in the previous paragraphs: even in nonlinear problems, IGA is extremely efficient for polynomial degrees up to 6 or 8, but fails to achieve spectral accuracy in the context of strong  $p$  or  $k$ -refinement. SEMs, although not competitive with IGA for lower degree polynomials, become the only viable option for polynomial degrees higher than 10.

For reference, we report the condition number of all matrices, both in the linear and in the nonlinear case, as a function of the polynomial degree and as a function of the number of degrees of freedom in Figure 14. While this comparison is not entirely fair (since the size of the IGA matrices is larger than their spectral counterpart), it is still significant to show the very large rate of growth of the condition number for IGA methods with the polynomial degree.

## 5. Conclusions

We presented some high-order numerical methods based on NURBS mappings, and applied them to the Laplace–Beltrami equation on some moderately complex surface geometries.

Founding a numerical method on NURBS maps allows to exactly represent the domain geometry and avoids the meshing construction process, a complex and time-consuming step usually done through isoparametric or transfinite maps in  $hp$ -Finite Element or Spectral Element Methods.

All the numerical methods considered achieve very fast convergence for

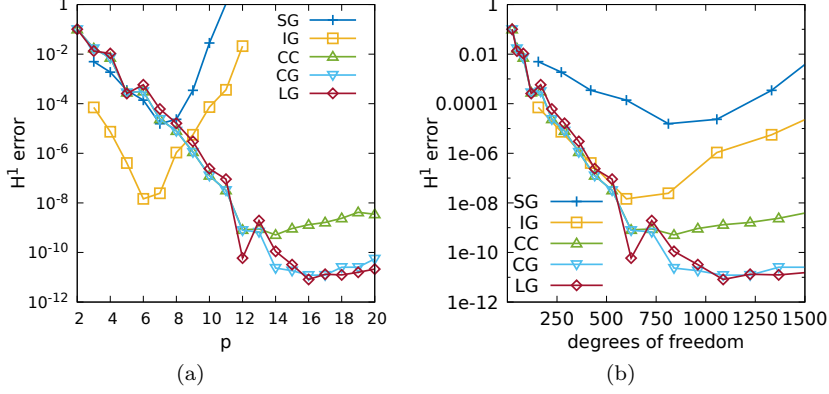


Figure 13: Error after termination of the fixed-point scheme for the Allen–Cahn equation, measured in the  $H^1$  norm, as a function of the polynomial degree  $p$  (left) and of the estimated operation count vs number of degrees of freedom (right). The figures refer to the annular geometry with forcing term chosen so that the exact solution is  $u(\mathbf{x}) = x_1^2 - x_2^3$ .

polynomial degrees between 3 and 8, but only the methods based on Lagrange interpolants at Gauss–Lobatto points reach consistently spectral precision.

The numerical evidence produced in this article raises some limitations to Isogeometric methods in the context of strong  $p$ -refinement. In particular, Isogeometric methods perform very well up to a polynomial degree between 8 and 10, then the stiffness matrix becomes too ill-conditioned and starts affecting accuracy. One notable exception is the Isogeometric Collocation method with globally  $C^1$  basis functions. This method has reached almost spectral accuracy in all of our numerical tests, at a computational cost comparable with that of the best performer.

Chebyshev multipatch collocation methods composed with NURBS maps seem to be a very good option to achieve high-order and high-precision approximations, avoiding at the same time the need for complex meshing subroutines, at least on two-dimensional problems. The extension to three-dimensional domains would require addressing the many inter-element patching conditions, that would add complexity to the implementation of this method. An important limitation of Chebyshev and Legendre methods is found in singular meshes. If a side of a quad collapses to a point, the Chebyshev and Legendre methods break due to the loss of degrees of freedom, and to the consequent singularity of the system’s matrix. In such a case, B-spline and NURBS methods are more robust and can still provide a solution.

A limitation of Chebyshev methods is that integration by parts can not be carried out due to the presence of a weighted inner product. This can be an issue when higher order differential operators are considered.

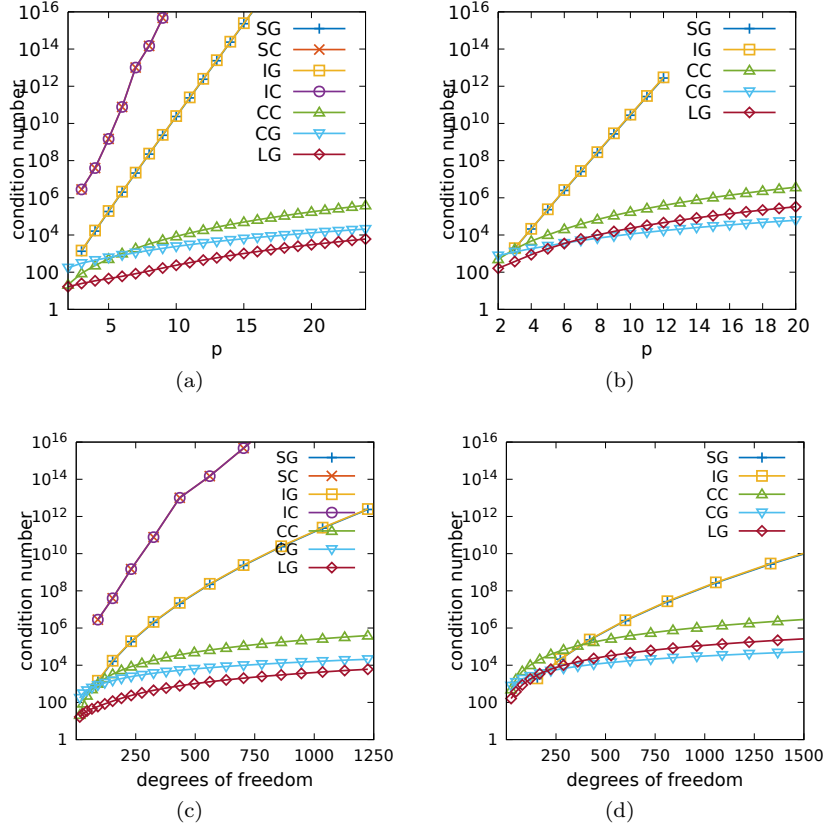


Figure 14: Matrix condition numbers for the  $k$ -refinement test on the annular surface. On the left, the results refer to the Laplace–Beltrami equation, while on the right figure the results refer to the fixed-point iteration for the Allen–Cahn equation. In the top row, the condition number is plot as a function of the polynomial degree  $p$ , while in the bottom row the condition number is shown as a function of the number of degrees of freedom.

## References

- [1] Anitescu, C., Jia, Y., Zhang, Y. J., Rabczuk, T., 2015. An isogeometric collocation method using superconvergent points. *Computer Methods in Applied Mechanics and Engineering* 284, 1073–1097.
- [2] Auricchio, F., ao da Veiga, L. B., Hughes, T., Reali, A., Sangalli, G., 2012. Isogeometric collocation for elastostatics and explicit dynamics. *Computer Methods in Applied Mechanics and Engineering* 249-252, 2–14.
- [3] Auricchio, F., Calabrò, F., Hughes, T., Reali, A., Sangalli, G., dec 2012. A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 249-252 (January), 15–27.
- [4] Auricchio, F., Da Veiga, L. B., Hughes, T. J. R., Reali, A., Sangalli, G., 2010. Isogeometric Collocation Methods. *Mathematical Models and Methods in Applied Sciences* 20 (11), 2075–2107.
- [5] Bartezzaghi, A., Dedè, L., Quarteroni, A., 2015. Isogeometric analysis of high order partial differential equations on surfaces. *Computer Methods in Applied Mechanics and Engineering* 295, 446–469.
- [6] Bartezzaghi, A., Dedè, L., Quarteroni, A., 2016. Isogeometric analysis of geometric partial differential equations. *Computer Methods in Applied Mechanics and Engineering* 311, 625–647.
- [7] Bazilevs, Y., Beirão Da Veiga, L., Cottrell, J. A., Hughes, T. J. R., Sangalli, G., jul 2006. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences* 16 (07), 1031–1090.
- [8] Beirão da Veiga, L., Buffa, A., Rivas, J., Sangalli, G., 2011. Some estimates for h–p–k-refinement in isogeometric analysis. *Numerische Mathematik* 118 (2), 271–305.
- [9] Benson, D. J., Bazilevs, Y., Hsu, M. C., Hughes, T. J. R., 2010. Isogeometric shell analysis: The Reissner-Mindlin shell. *Computer Methods in Applied Mechanics and Engineering* 199, 276 – 289.
- [10] Boyd, J. P., 2001. *Chebyshev and Fourier Spectral Methods*. Dover.
- [11] Calabrò, F., Sangalli, G., Tani, M., 2017. Fast formation of isogeometric galerkin matrices by weighted quadrature. *Computer Methods in Applied Mechanics and Engineering* 316, 606 – 622, special Issue on Isogeometric Analysis: Progress and Challenges.

- [12] Canuto, C., Hussaini, M., Quarteroni, A., Zang, T., 2006. Spectral Methods, Fundamentals in Single Domains. Springer-Verlag.
- [13] Canuto, C., Hussaini, M., Quarteroni, A., Zang, T., 2007. Spectral Methods, Evolution to Complex Geometries and Applications to Fluid Dynamics. Springer-Verlag.
- [14] Cirak, F., Ortiz, M., Schröder, P., 2000. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering* 47 (12), 2039–2072.
- [15] Cottrell, J., Hughes, T., Bazilevs, Y., 2009. Isogeometric Analysis: Toward Integration of CAD and FEA. Wiley.
- [16] Cottrell, J. A., Reali, A., Bazilevs, Y., Hughes, T. J. R., 2006. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering* 195 (41-43), 5257–5296.
- [17] Dalcin, L., Collier, N., Vignal, P., Cortes, A., Calo, V., 2016. Petiga: A framework for high-performance isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 308, 151–181.
- [18] Dedè, L., Quarteroni, A., 2015. Isogeometric analysis for second order partial differential equations on surfaces. *Computer Methods in Applied Mechanics and Engineering* 284, 807–834.
- [19] Demko, S., 1985. On the existence of interpolating projections onto spline spaces. *Journal of Approximation Theory* 43 (2), 151 – 156.
- [20] Deville, M., Fischer, P., Mund, E., 2002. High-Order Methods for Incompressible Fluid Flow. Cambridge.
- [21] Dornisch, W., Klinkel, S., 2014. Treatment of Reissner-Mindlin shells with kinks without the need for drilling rotation stabilization in an isogeometric framework. *Computer Methods in Applied Mechanics and Engineering* 276, 35–66.
- [22] Dornisch, W., Klinkel, S., Simeon, B., 2013. Isogeometric Reissner-Mindlin shell analysis with exactly calculated director vectors. *Computer Methods in Applied Mechanics and Engineering* 253, 491–504.
- [23] Gottlieb, D., Orszag, S. A., 1977. Numerical Analysis of Spectral Methods. SIAM.
- [24] Heltai, L., Arroyo, M., DeSimone, A., jan 2014. Nonsingular isogeometric boundary element method for Stokes flows in 3D. *Computer Methods in Applied Mechanics and Engineering* 268, 514–539.

- [25] Heltai, L., Kiendl, J., DeSimone, A., Reali, A., apr 2017. A natural framework for isogeometric fluid-structure interaction based on BEM-shell coupling. *Computer Methods in Applied Mechanics and Engineering* 316, 522–546.
- [26] Hughes, T., Cottrell, J., Bazilevs, Y., 2005. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 194 (39-41), 4135–4195.
- [27] Hughes, T., Reali, A., Sangalli, G., jan 2010. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 199 (5-8), 301–313.
- [28] Jones, E., Oliphant, T., Peterson, P., et al., 2001–. SciPy: Open source scientific tools for Python. [Online].
- [29] Kiendl, J., Bletzinger, K.-U., Linhard, J., Wüchner, R., 2009. Isogeometric shell analysis with Kirchhoff-Love elements. *Computer Methods in Applied Mechanics and Engineering* 198, 3902–3914.
- [30] Kiendl, J., Hsu, M.-C., Wu, M., Reali, A., 2015. Isogeometric Kirchhoff-Love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering* 291, 280–303.
- [31] Klingenberg, W., 1978. *A Course in Differential Geometry*. Springer-Verlag.
- [32] Lablée, O., 2015. *Spectral Theory in Riemannian Geometry*. European Mathematical Society.
- [33] Lee, J. M., 1997. *Riemannian Manifolds*. Springer-Verlag.
- [34] Lorenzis, L. D., Evans, J., Hughes, T., Reali, A., 2015. Isogeometric collocation: Neumann boundary conditions and contact. *Computer Methods in Applied Mechanics and Engineering* 284, 21 – 54, isogeometric Analysis Special Issue.
- [35] Manni, C., Sablonnière, P., 2007. Quadratic spline quasi-interpolants on powell-sabin partitions. *Advances in Computational Mathematics* 26 (1), 283–304.
- [36] Piegl, L., Tiller, W., 1995. *The NURBS Book*. Springer-Verlag.
- [37] Rosenberg, S., 1997. *The Laplacian on a Riemannian Manifold*. London Mathematical Society.
- [38] Rotundo, N., Kim, T.-Y., Jiang, W., Heltai, L., Fried, E., 2016. Error analysis of a b-spline based finite-element method for modeling wind-driven ocean circulation. *Journal of Scientific Computing* 69 (1), 430–459.



- [39] Schillinger, D., Evans, J. A., Reali, A., Scott, M. A., Hughes, T. J., 2013. Isogeometric collocation: Cost comparison with galerkin methods and extension to adaptive hierarchical NURBS discretizations. *Computer Methods in Applied Mechanics and Engineering* 267, 170–232.
- [40] Schillinger, D., Hossain, S. J., Hughes, T. J., aug 2014. Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 277, 1–45.
- [41] Sevilla, R., Fernández-Méndez, S., Huerta, A., 2008. Nurbs-enhanced finite element method (nefem). *International Journal for Numerical Methods in Engineering* 76 (1), 56–83.
- [42] Speleers, H., Manni, C., Pelosi, F., Sampoli, M. L., 2012. Isogeometric analysis with powell–sabin splines for advection-diffusion-reaction problems. *Computer Methods in Applied Mechanics and Engineering* 221-222, 132 – 148.
- [43] Uhm, T.-K., Youn, S.-K., 2009. T-spline finite element method for the analysis of shell structures. *International Journal for Numerical Methods in Engineering* 80, 507–536.
- [44] van der Walt, S., Colbert, S. C., Varoquaux, G., 2011. The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering* 13 (2), 22–30.